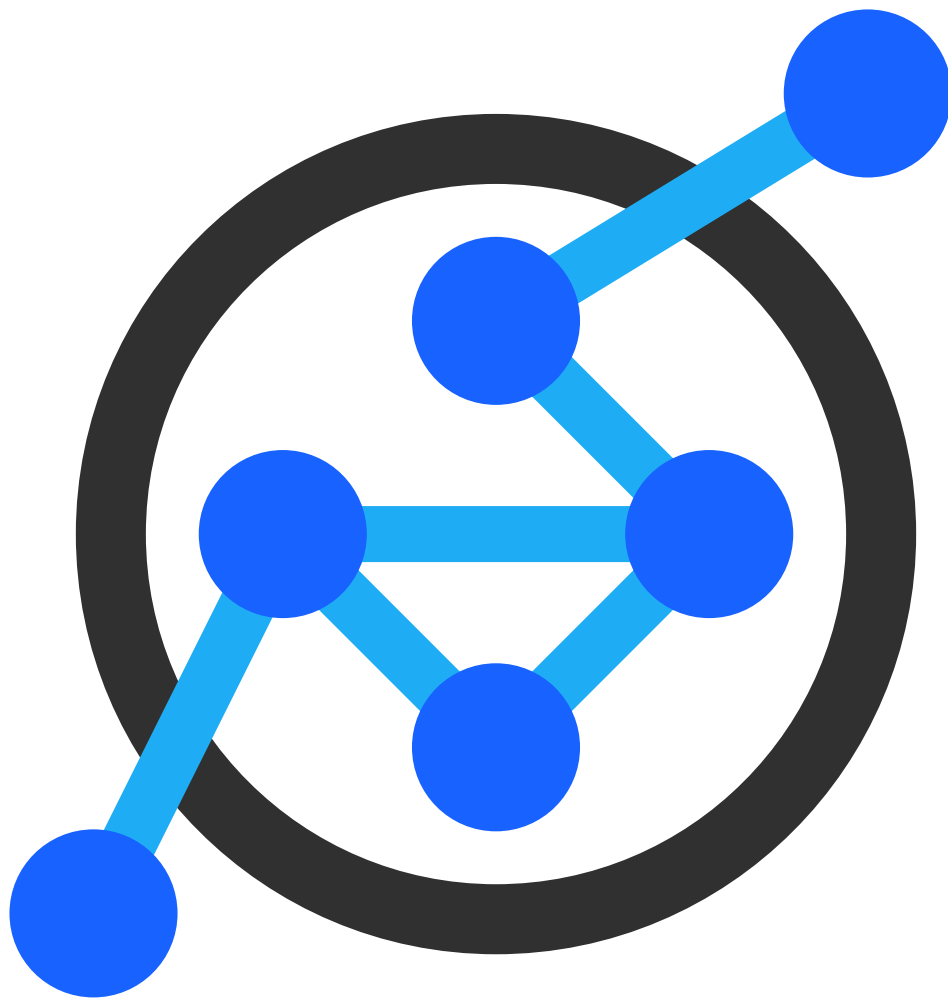


Round 1 SOI 2026/2027

Recueil de Solutions



Swiss Olympiad in Informatics
01 September 2026 – 25 September 2026



Piles de livres

Préparation de la tâche

Mark Neumann

Traduction (de)

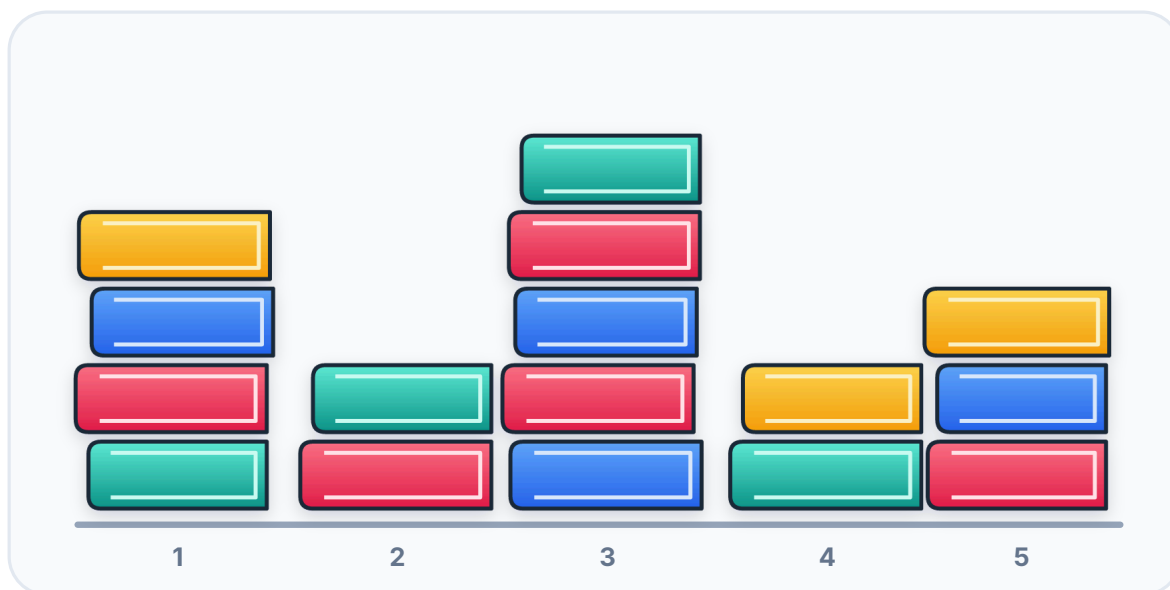
Benjamin Schmid

Traduction (fr)

Cheng Zhong

Auteur de la solution

Benjamin Schmid



Partie 1: Pour ses examens de maturité, la souris Stofl doit lire 5 piles de livres, comme illustré sur l'image. Il constate que les cinq piles de livres n'ont pas la même hauteur, ce qui le dérange. Quel est le nombre minimal de livres que la souris Stofl devrait ajouter pour amener toutes les piles à la même hauteur ? Il ne peut retirer aucun livre, car il doit tous les lire.

- ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Partie 2: La souris Stofl se rend compte qu'il peut aussi déplacer des livres, un à la fois, d'une pile vers n'importe quelle autre pile. Quel est le nombre minimal de livres qu'il faudrait déplacer ou ajouter pour amener toutes les piles à la même hauteur ?

- ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6



Sous-tâche 1: Ajouter des livres

Nous devons ajouter 9 livres.

Dans cette partie, nous pouvons uniquement ajouter des livres. Nous ne pouvons ni en retirer ni en déplacer. Une pile ne sera donc jamais moins haute qu'au départ. Le mieux que nous puissions faire est alors de trouver la pile la plus haute et d'ajouter des livres à toutes les autres piles pour qu'elles atteignent cette hauteur.

La pile la plus haute est la pile 3, avec 5 livres. Il manque 1 livre à la pile 1, 3 livres à la pile 2, 3 livres à la pile 4 et 2 livres à la pile 5. Au total, nous devons donc ajouter 9 livres.

Sous-tâche 2: Déplacer des livres

Nous devons effectuer 5 actions.

Dans la deuxième partie, nous pouvons aussi déplacer des livres. Comme nous l'avons vu dans la première partie, le nombre de livres à ajouter est déterminé par la pile la plus haute. Nous souhaitons donc que cette pile soit aussi basse que possible. Pour cela, nous répartissons les livres aussi uniformément que possible entre les piles. La pile la plus haute après cette répartition nous indique alors la hauteur que chaque pile devra atteindre. Notons que, si nous répartissons les livres de cette manière, la pile la plus haute dépasse la pile la plus basse d'au plus un livre.

Nous avons 16 livres au total. Avec seulement 3 livres par pile, nous ne pouvons avoir que 15 livres. Il nous faut donc 4 livres par pile, ce qui permettrait d'avoir 20 livres au total. Pour y parvenir, nous pouvons par exemple déplacer un livre de la pile 3 vers la pile 5, puis ajouter deux livres à chacune des piles 2 et 4. Cela fait 5 actions au total.



Plus court chemin

Préparation de la tâche

Mark Neumann

Traduction (de)

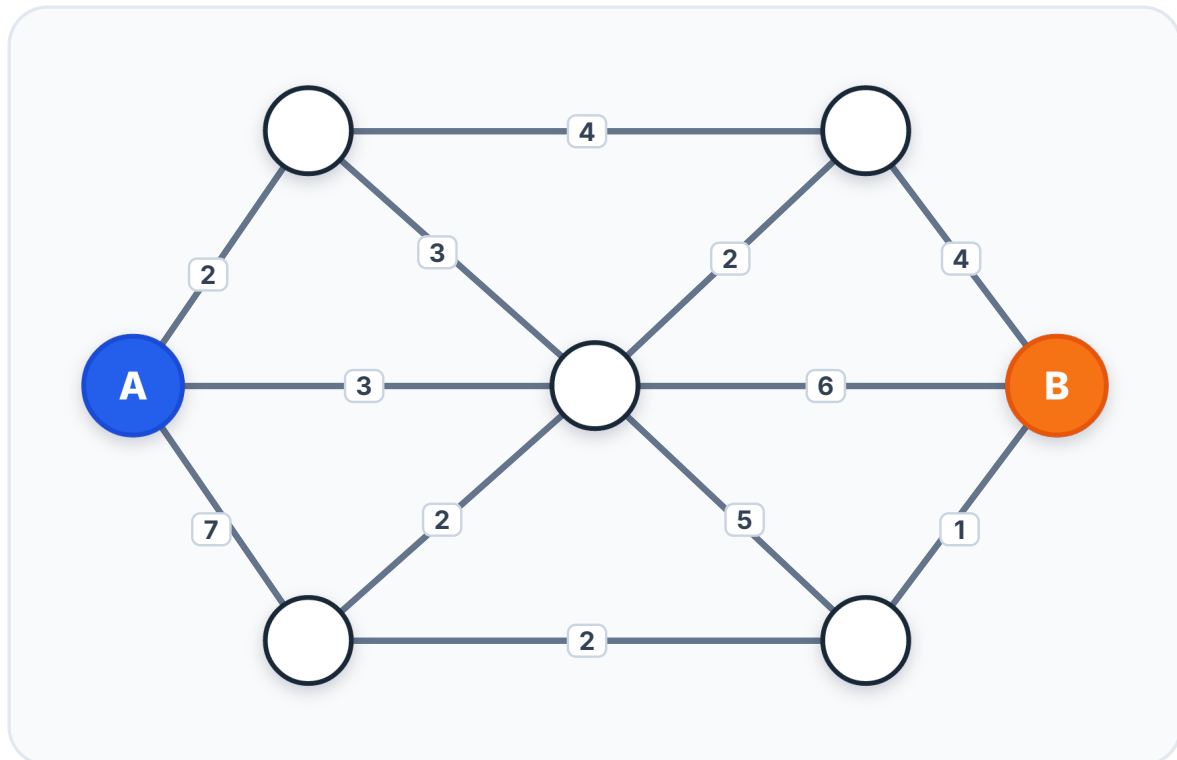
Benjamin Schmid

Traduction (fr)

Cheng Zhong

Auteur de la solution

Benjamin Schmid



Partie 1: La souris Binna vit actuellement dans la ville A et doit se rendre le plus rapidement possible dans la ville B. Certaines villes sont reliées par une route, chacune pouvant être empruntée dans les deux directions. Pour chaque route, nous savons combien de temps il faut pour la parcourir. L'image montre une carte du réseau de villes et de routes (avec les temps de trajet correspondants). Quel est le temps de trajet le plus court possible entre la ville A et la ville B ?

- ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11

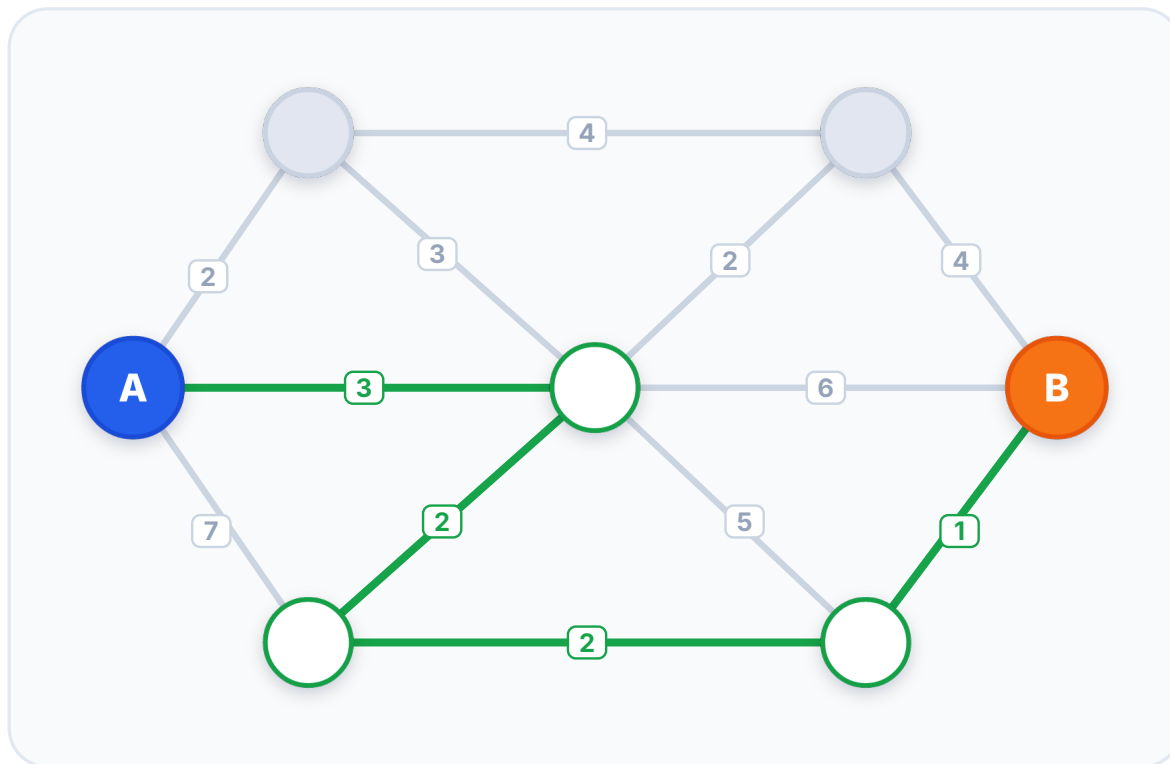
Partie 2: En raison d'une tempête, au plus une des routes peut être fermée. Quel est le temps minimal que la souris Binna doit prévoir pour que, quelle que soit la route fermée, il existe un chemin lui permettant d'aller de la ville A à la ville B dans ce temps ? Notez que Binna saura quelle route est fermée avant de commencer son trajet.

- ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12



Sous-tâche 1: Trajet direct

Le temps de trajet le plus court possible est de 8.



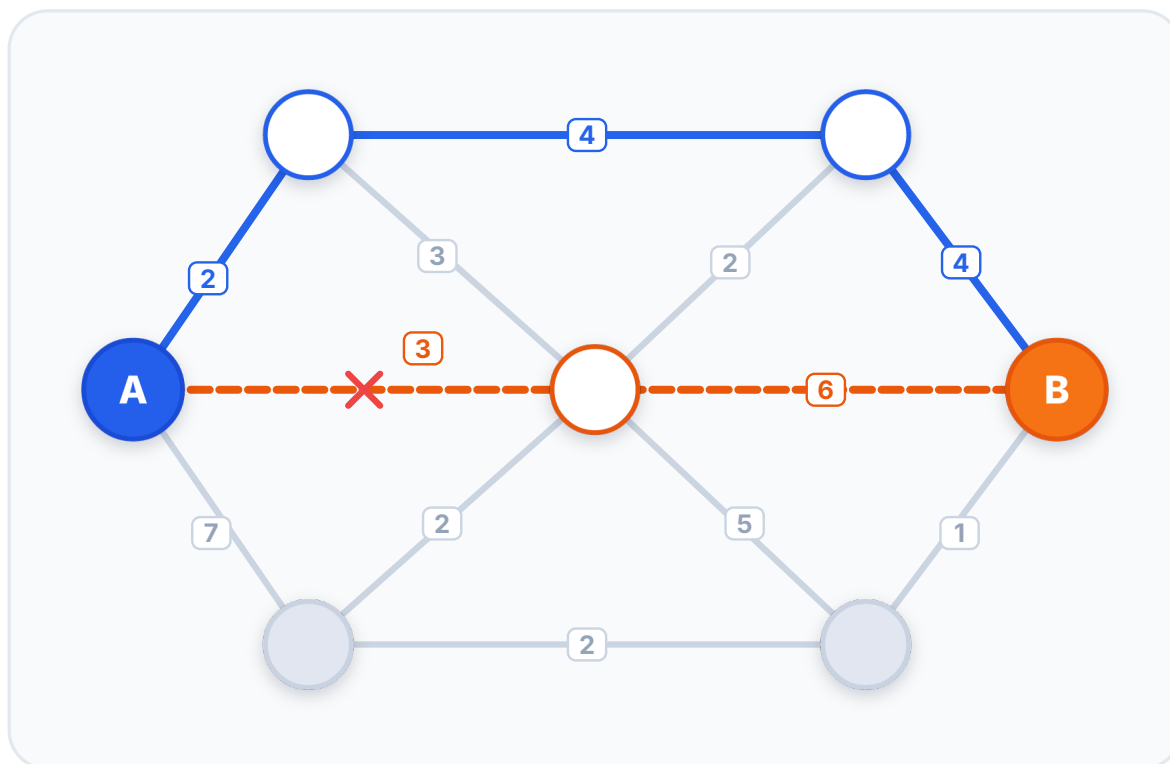
En informatique, ce type de réseau constitué de nœuds (= villes) et d'arêtes (= routes) s'appelle un graphe. Un graphe représente des entités (les nœuds, ici les villes) et leurs connexions (les arêtes, ici les routes). Une arête peut avoir un poids qui indique une sorte de « distance » associée à cette connexion.

Plusieurs algorithmes permettent de trouver le plus court chemin dans un graphe ; le plus connu est l'algorithme de Dijkstra.



Sous-tâche 2: Routes fermées

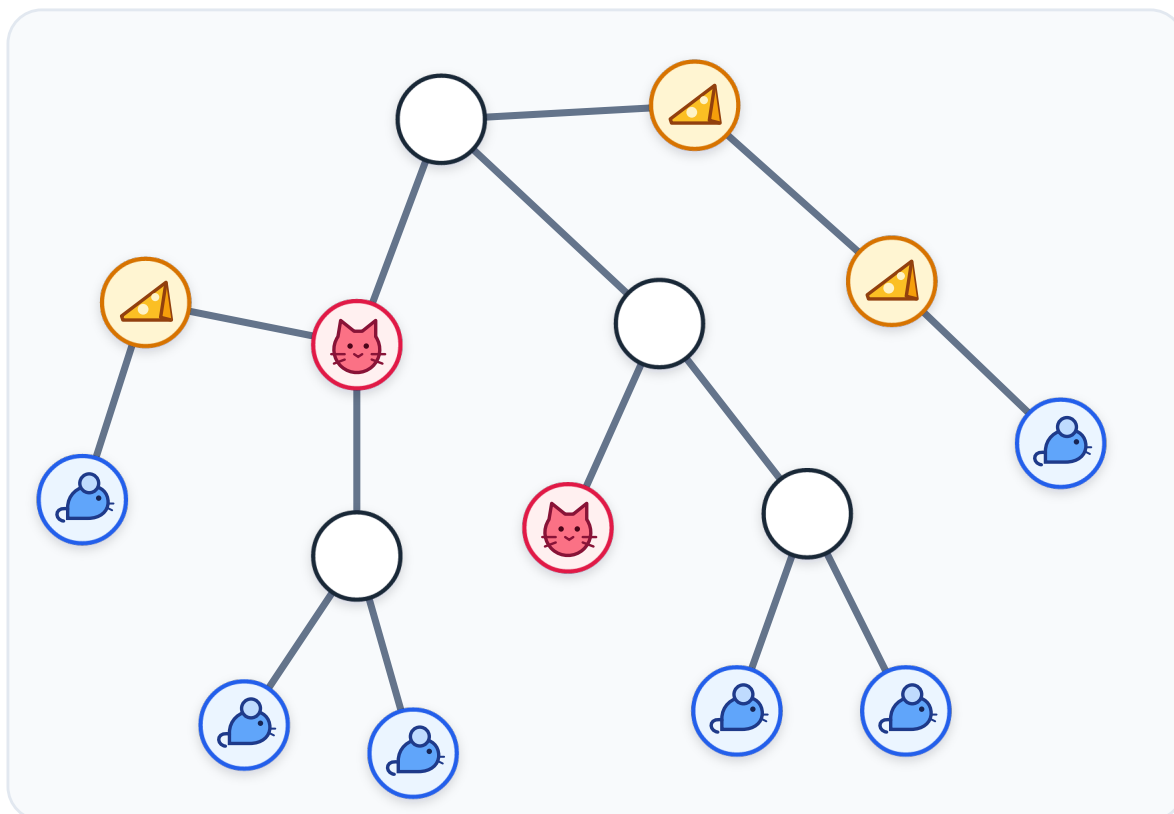
La souris Binna doit prévoir un temps de trajet d'au moins 10.



Comme la souris Binna connaît à l'avance la route fermée, nous pouvons simuler les différents cas selon la route concernée. Une première observation est que, si la route fermée ne se trouve pas sur le plus court chemin de la partie 1, le temps de trajet de Binna n'augmente pas. Nous devons donc uniquement considérer les cas où l'une des routes du plus court chemin est indisponible. Nous pouvons retirer à tour de rôle chacune des arêtes de ce chemin et recalculer le plus court trajet. La solution est le plus grand des temps de trajet ainsi obtenus.

Si nous retirons l'arête de longueur 3 à gauche (barrée sur l'image ci-dessus), le plus court chemin obtenu a une longueur de 10 et emprunte toujours les arêtes situées tout en haut (chemin bleu). Si nous retirons n'importe quelle autre arête du plus court chemin initial, nous pouvons toujours emprunter le chemin qui va tout droit, de longueur 9 (chemin orange).

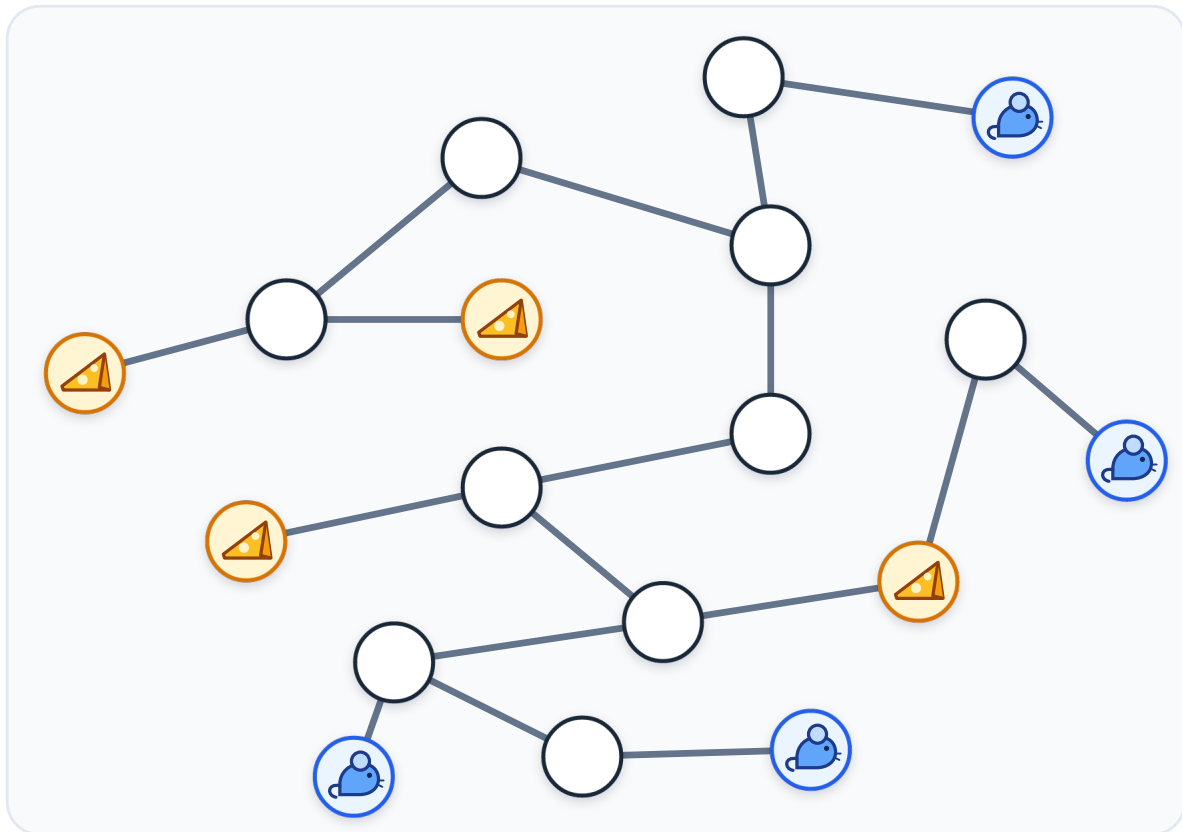
Mark Neumann
Benjamin Schmid
Cheng Zhong
Benjamin Schmid



Partie 1: Vous venez de construire un labyrinthe à souris. Le labyrinthe est composé de différentes salles reliées par des couloirs, comme illustré sur l'image.

Certaines salles du labyrinthe contiennent des souris, tandis que d'autres contiennent des morceaux de fromage. Certaines salles sont gardées par un chat, que les souris ne peuvent pas franchir. Dans la configuration illustrée, combien existe-t-il de paires (souris, fromage) différentes telles que la souris puisse atteindre le fromage sans passer par une salle gardée par un chat ? Une même souris peut être associée à plusieurs morceaux de fromage, et inversement.

- ☐ 4 ☐ 6 ☐ 7 ☐ 10 ☐ 12



Partie 2: L'heure du goûter est terminée, et vous voulez maintenant empêcher les souris d'atteindre le fromage. Le labyrinthe ressemble désormais à celui de l'image.

Quel est le nombre minimal de salles dans lesquelles vous devez placer un chat pour qu'aucune souris ne puisse atteindre un morceau de fromage ? Vous ne pouvez pas placer de chat dans une salle contenant une souris ou du fromage.

- ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6



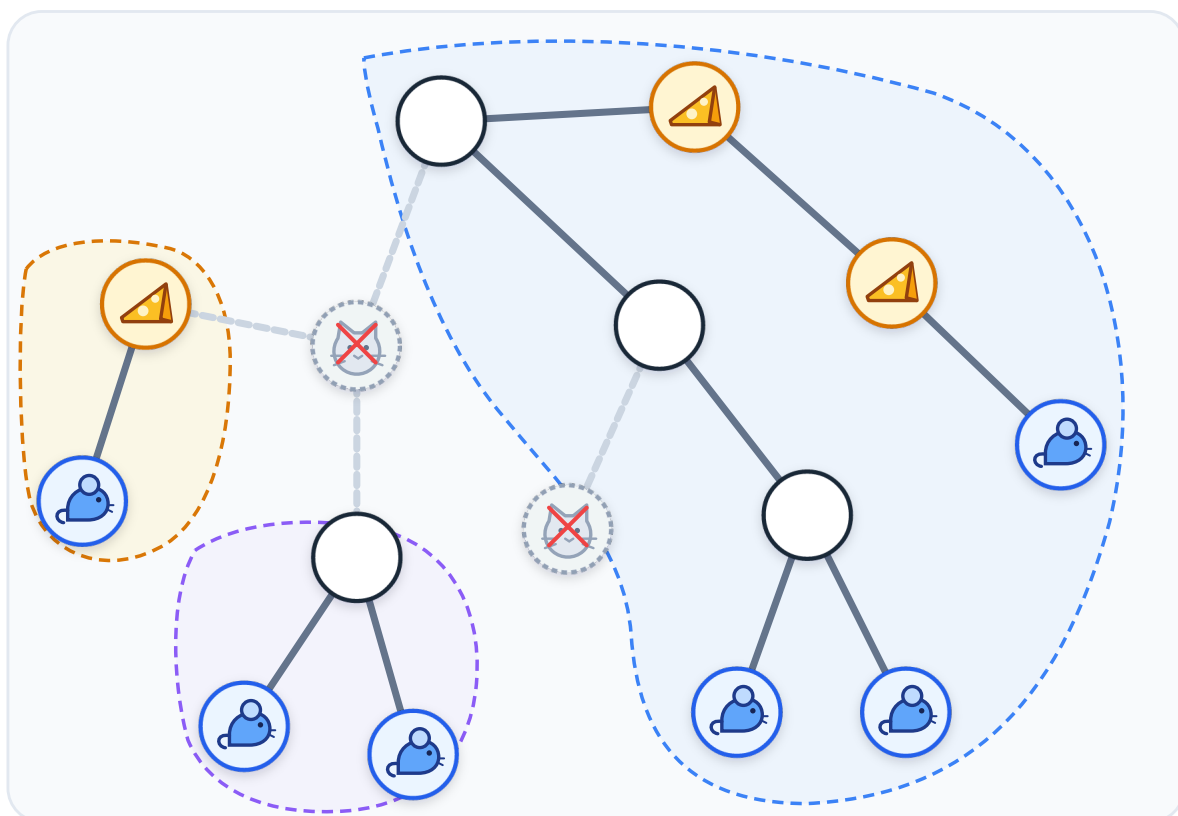
Sous-tâche 1: Compter les paires

Il y a 7 paires (souris, fromage) pour lesquelles la souris peut atteindre le fromage.

Cette représentation du labyrinthe à l'aide de nœuds (= salles) et d'arêtes (= couloirs) s'appelle un graphe. Dans la première partie, nous nous intéressons aux composantes connexes du graphe. Une composante connexe est un ensemble de nœuds du graphe reliés entre eux par des arêtes, directement ou indirectement.

Au départ, il n'y a qu'une seule composante : si nous ignorons les chats, nous pouvons atteindre n'importe quel nœud depuis n'importe quel autre. Cependant, nous savons qu'une souris ne peut pas passer par un nœud occupé par un chat. Nous pouvons donc retirer ces nœuds du graphe, ce qui le sépare en plusieurs composantes. Une souris située dans une composante peut atteindre tous les morceaux de fromage de cette même composante, mais aucun de ceux d'une autre composante. Nous pouvons donc examiner chaque composante séparément et multiplier son nombre de souris par son nombre de morceaux de fromage pour obtenir le nombre de paires qu'elle contient. Il suffit ensuite d'additionner les résultats de toutes les composantes pour obtenir le total.

Le résultat est donc $1 \cdot 1 + 2 \cdot 0 + 3 \cdot 2 = 1 + 0 + 6 = 7$.



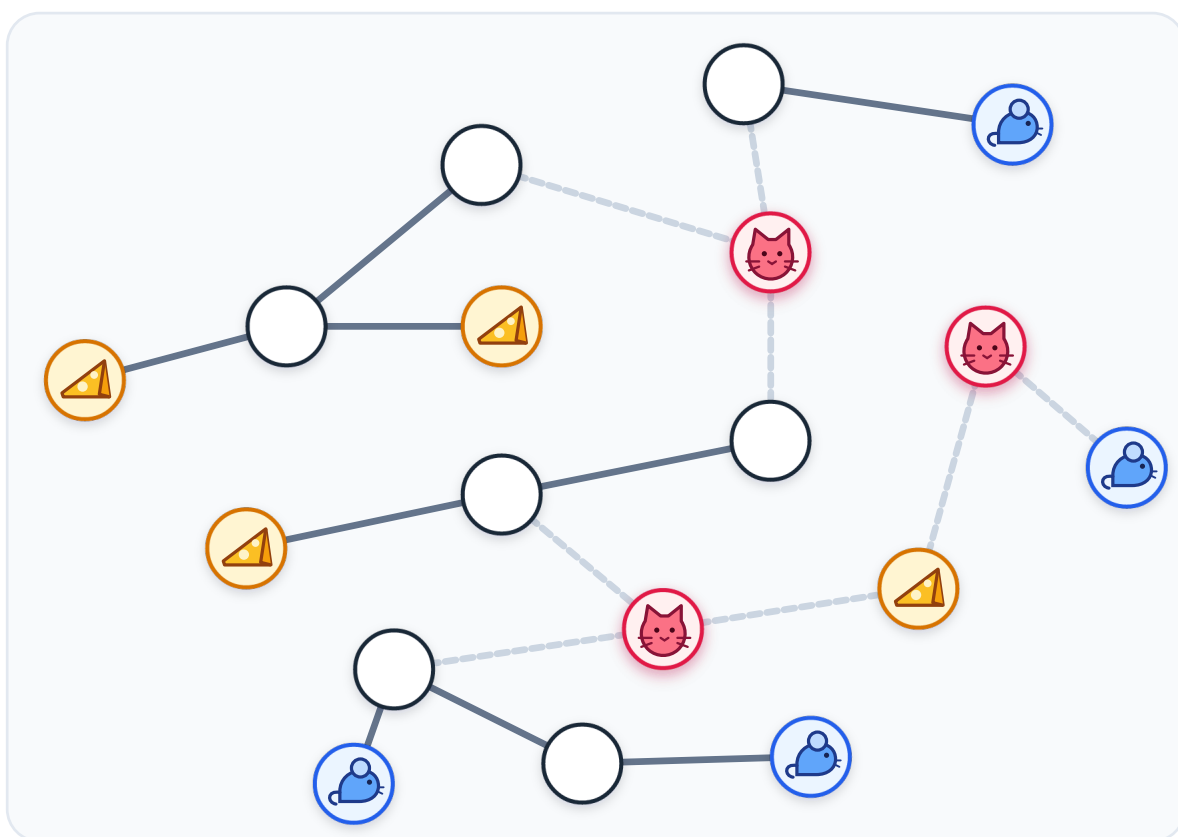
Les trois composantes sont mises en évidence par des couleurs différentes : celle de droite (en bleu) contient trois souris et deux morceaux de fromage ($3 \cdot 2 = 6$ paires), celle de gauche (en jaune) contient une souris et un morceau de fromage ($1 \cdot 1 = 1$ paire), et celle du bas (en violet) contient deux souris, mais aucun fromage ($2 \cdot 0 = 0$ paire).



Sous-tâche 2: L'heure du goûter est terminée

Il nous faut au moins 3 chats.

Dans la deuxième partie, nous devons choisir les nœuds où placer des chats pour que les composantes obtenues contiennent uniquement des souris ou uniquement du fromage, mais pas les deux. Il existe plusieurs possibilités ; l'une d'elles est illustrée ci-dessous.



Les trois chats sont placés dans les salles indiquées en rouge sur le schéma. Cela sépare le labyrinthe en six composantes contenant chacune uniquement des souris ou uniquement du fromage.

Pour un problème de si petite taille, nous pouvons trouver une solution à la main en essayant quelques nœuds différents. Une solution générale pourrait utiliser un algorithme de flot maximal pour trouver une coupe minimale dans un graphe construit spécialement à cet effet. L'explication de cette méthode dépasse le cadre de ce livret de solutions.



Randonnée

Préparation de la tâche

Mark Neumann, Benjamin Schmid

Traduction (de)

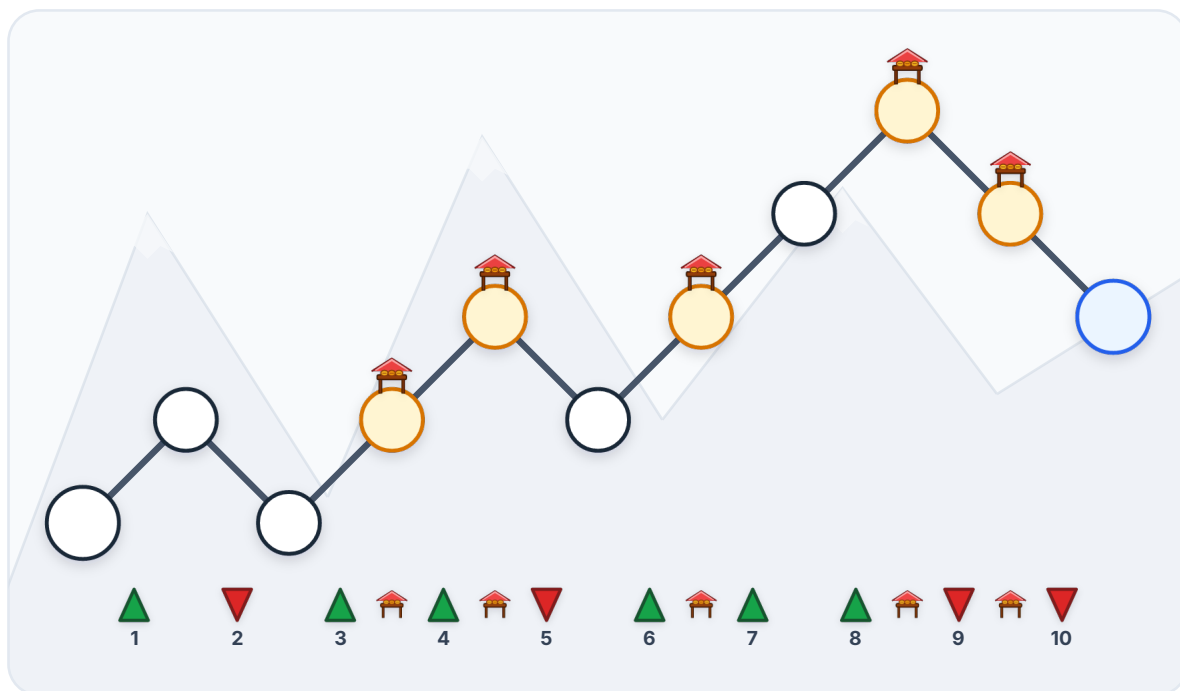
Benjamin Schmid

Traduction (fr)

Cheng Zhong

Auteur de la solution

Benjamin Schmid



Partie 1: Vous voulez faire une randonnée en montagne et emportez 3 sandwichs avec vous.

La randonnée est composée de 10 tronçons et passe devant quelques kiosques à pain, comme illustré sur l'image. Pour chaque tronçon, vous suivez la séquence suivante en deux étapes avant de commencer à marcher :

Regardez le point de départ actuel du tronçon.

- S'il y a un kiosque : vous achetez 2 sandwichs.
- S'il n'y a pas de kiosque : vous achetez 0 sandwich.

Regardez le tronçon que vous vous apprêtez à parcourir.

- Si le chemin monte et que vous êtes à un kiosque : vous mangez 2 sandwichs.
- Si le chemin monte et que vous n'êtes **pas** à un kiosque : vous mangez 1 sandwich.
- Si le chemin descend : vous mangez 0 sandwich.

Combien de sandwichs vous restera-t-il à la fin de la randonnée ?

Note : Si vous n'avez pas assez de sandwichs, vous devrez interrompre votre randonnée.

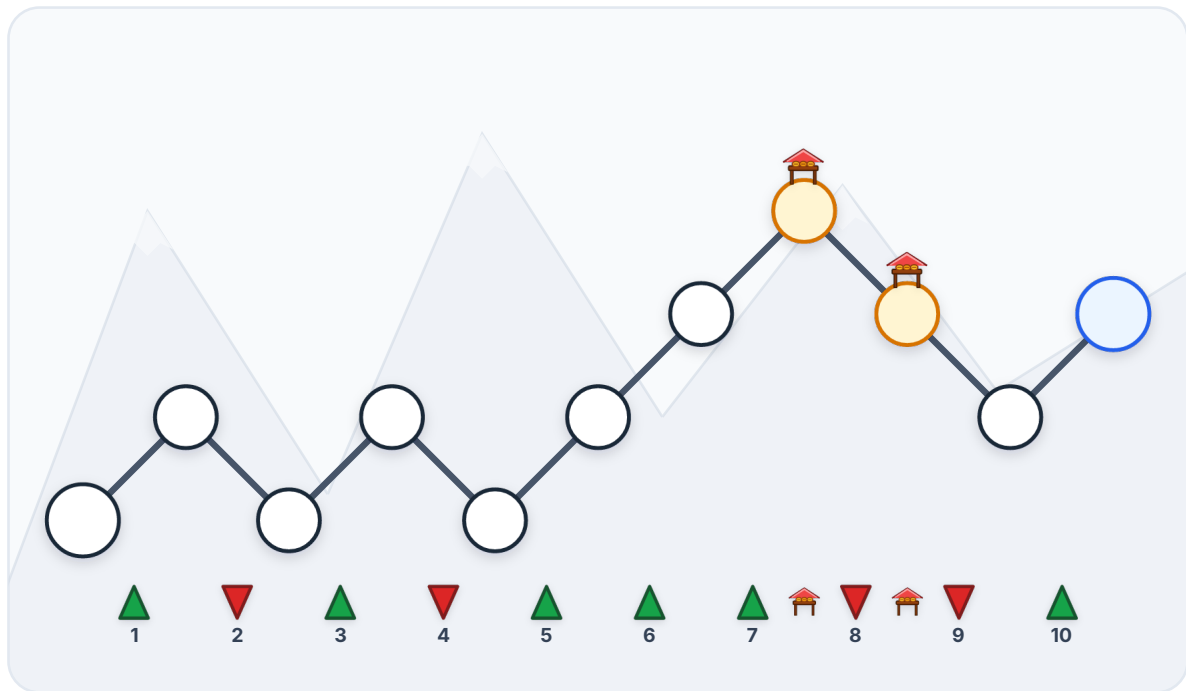
☐ 3

☐ 4

☐ 5

☐ 6

☐ 8



Partie 2: Une fois arrivé au point de départ, vous vous rendez compte que vous avez oublié vos sandwiches ! Il n'y a qu'une seule solution : espérer que de nouveaux kiosques ouvrent en chemin. De nouveaux kiosques peuvent être ajoutés au début de n'importe quel tronçon (y compris le premier) où il n'y a pas déjà un kiosque.

Quel est le nombre minimal de nouveaux kiosques qui doivent être ajoutés pour que vous puissiez terminer avec succès toute la randonnée en suivant les règles, bien que vous ayez commencé avec zéro sandwich ?

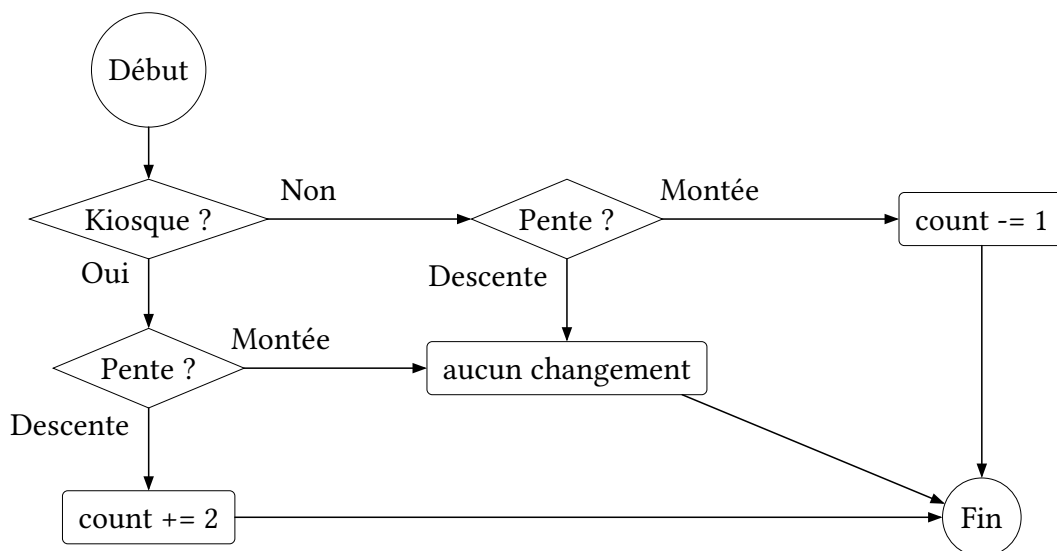
- ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4



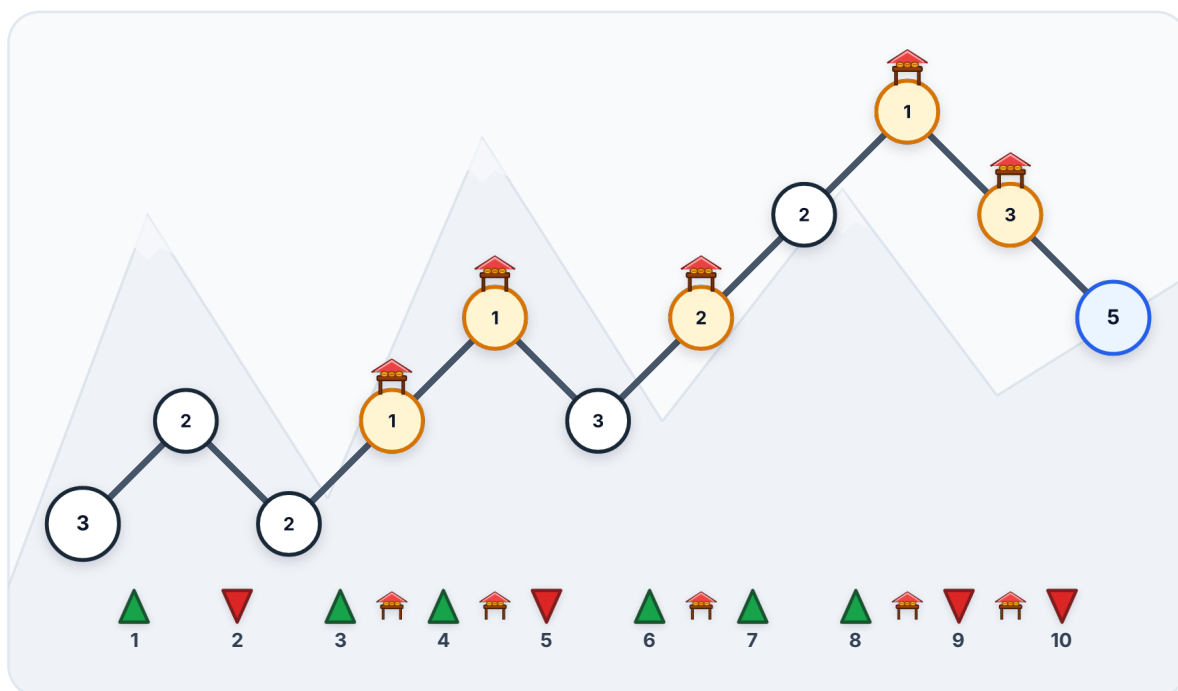
Sous-tâche 1: Trois sandwiches

Il nous reste 5 sandwiches à la fin de la randonnée.

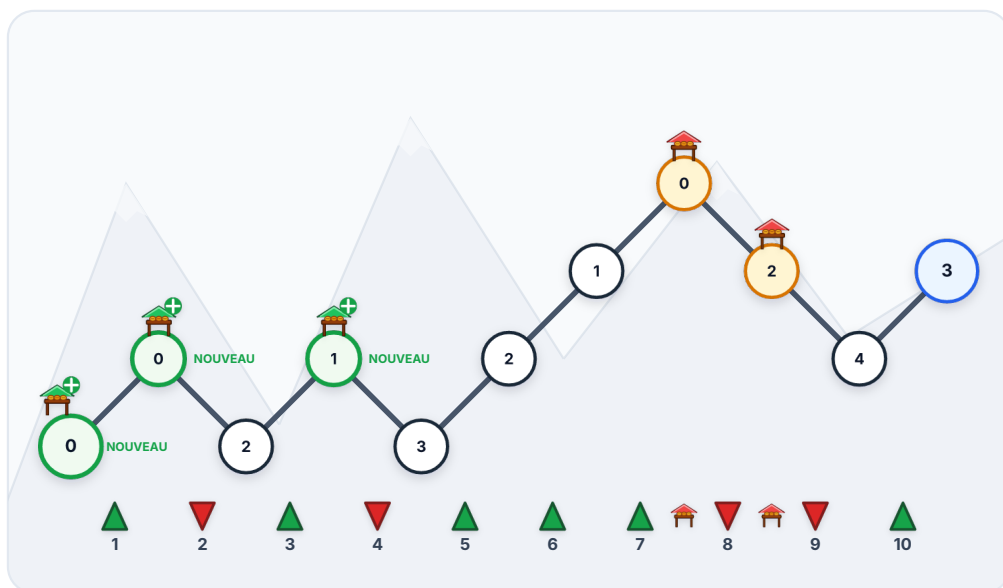
Cette tâche consiste à exécuter un algorithme simple et à en déterminer le résultat. Nous pouvons simplifier légèrement l'algorithme en notant simplement comment le nombre de sandwiches évolue selon la présence ou l'absence d'un kiosque et selon la pente. Nous initialisons le nombre de sandwiches (appelé « count » ci-dessous) à 3, puis nous exécutons les étapes suivantes pour chaque tronçon :



Nous obtenons les nombres de sandwiches suivants aux différents nœuds :



Nous devons ajouter au moins 3 kiosques.



Dans cette partie, nous devons déterminer combien de kiosques ajouter et où les placer. Nous pouvons faire quelques observations :

- Si nous plaçons un kiosque avant un tronçon en montée, nous pouvons le parcourir, mais nous aurons ensuite autant de sandwiches qu'avant. Ce kiosque permet donc de franchir une montée supplémentaire : celle qui le suit.
- Si nous plaçons un kiosque avant un tronçon en descente, nous aurons ensuite deux sandwiches de plus, qui permettent de franchir deux montées supplémentaires sans kiosque à leur départ.
- Il est préférable de placer un kiosque avant une descente plutôt qu'avant une montée : dans le premier cas, nous pouvons franchir deux montées, contre une seule dans le second.
- Nous ne devrions donc placer un kiosque avant une montée que si cela est absolument nécessaire.

Ces observations nous permettent de choisir les meilleurs emplacements pour les kiosques supplémentaires. Il en faut un avant le premier tronçon, car nous partons sans sandwich et ce tronçon monte. Avant le deuxième, nous vérifions si nous pouvons atteindre la fin sans autre ajout : ce n'est pas le cas. Nous y ajoutons donc aussi un kiosque, ce qui nous laisse deux sandwiches après ce tronçon. Le troisième tronçon monte à nouveau, mais nous avons des sandwiches, donc nous n'ajoutons pas de kiosque. Avant le quatrième, nous vérifions à nouveau si nous pouvons atteindre la fin sans autre ajout. Ce n'est toujours pas le cas : nous y ajoutons un kiosque et avons alors trois sandwiches après ce tronçon. Les tronçons suivants montent, mais nous avons juste assez de sandwiches pour tous les parcourir, sans ajouter de kiosque. Une fois au sommet, les deux kiosques situés avant les descentes suffisent pour atteindre la fin sans autre ajout. Il nous faut donc seulement trois kiosques supplémentaires.



Solaria

Préparation de la tâche

Mark Neumann

Traduction (de)

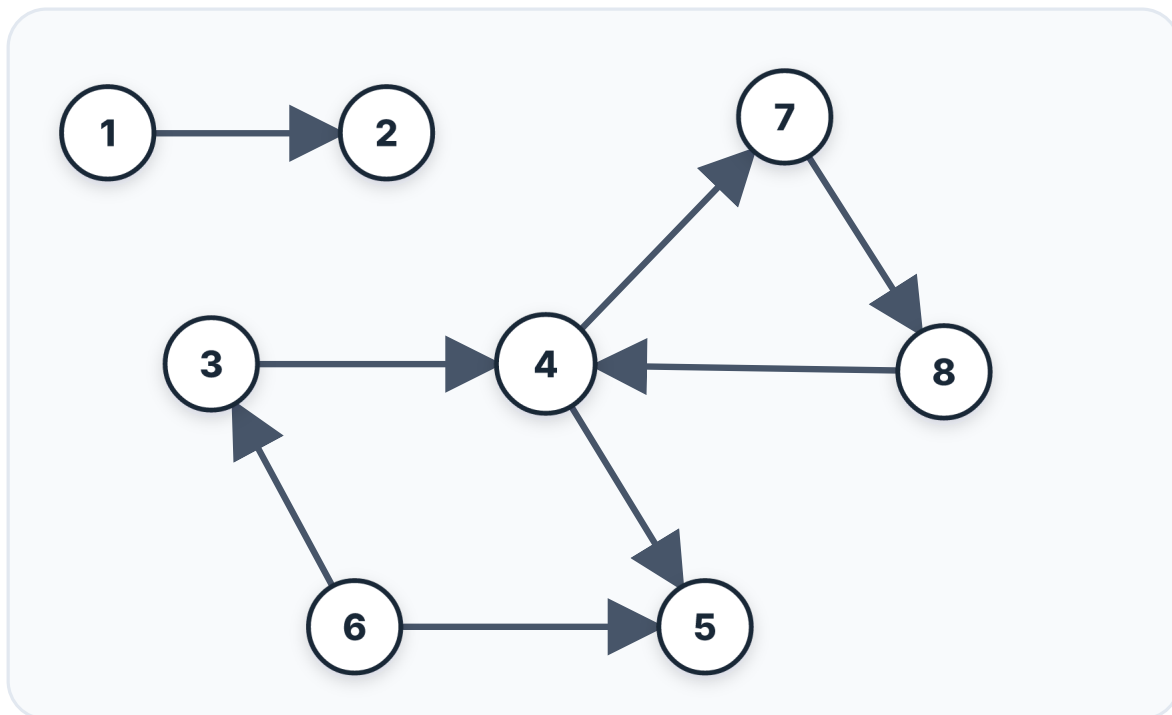
Benjamin Schmid

Traduction (fr)

Cheng Zhong

Auteur de la solution

Benjamin Schmid



Partie 1: La ville de montagne de Solaria dépend d'un réseau d'étroits ponts à sens unique pour relier ses quartiers. Mais les urbanistes ont commis un terrible oubli : comme les ponts ne peuvent être empruntés que dans un seul sens, les automobilistes se retrouvent piégés ! Il y a des parties de la ville d'où, une fois arrivé, aucun chemin ne permet de rentrer chez soi. Le maire veut réparer le réseau immédiatement, mais le budget de la ville ne permet de financer que quelques nouveaux ponts.

La carte du réseau de ponts sur l'image indique dans quel sens chaque pont peut être traversé. Quel est le nombre minimal de nouveaux ponts à sens unique que la ville doit construire pour qu'un automobiliste puisse partir de n'importe quel quartier et rejoindre avec succès tous les autres quartiers ?

Note : Il est autorisé que deux ponts relient les deux mêmes quartiers dans des sens opposés.

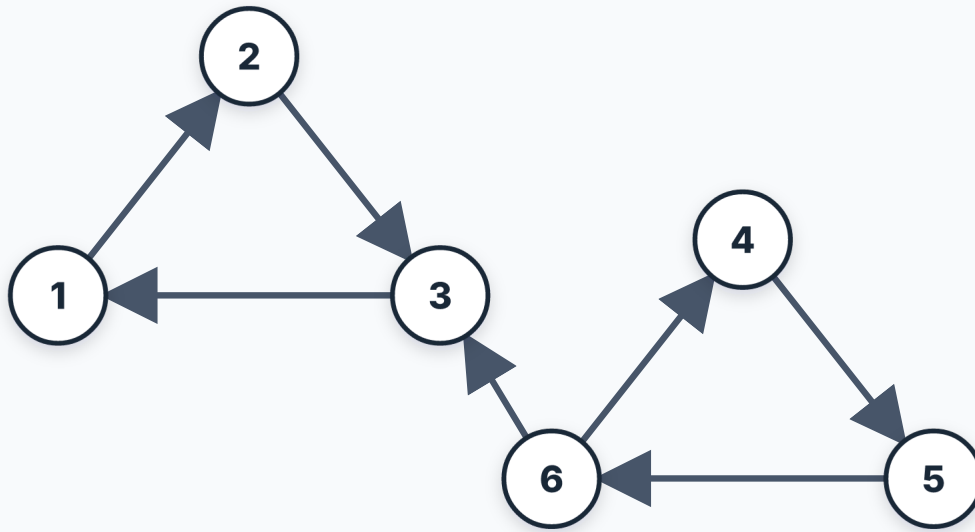
☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

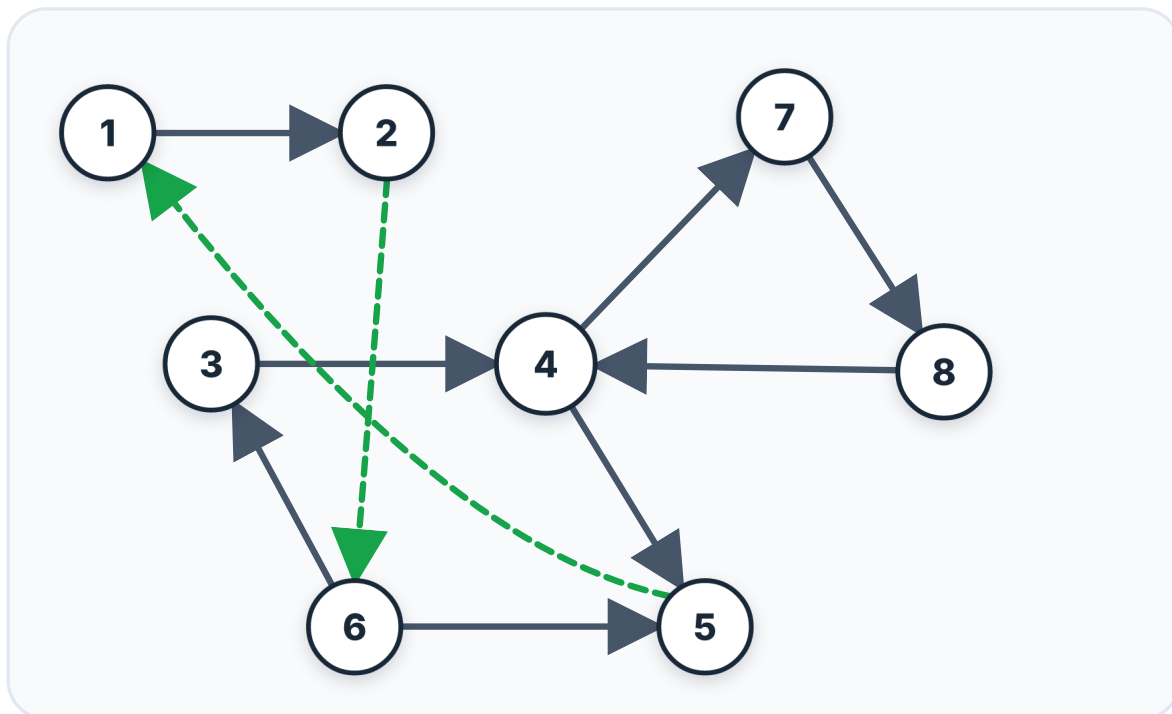


Partie 2: Considérez la nouvelle carte. Il y a plusieurs manières d'ajouter exactement un nouveau pont à sens unique afin qu'un automobiliste puisse partir de n'importe quel quartier et rejoindre avec succès tous les autres quartiers. De combien de manières différentes cela peut-il être fait ?

- ☐ 3 ☐ 5 ☐ 6 ☐ 8 ☐ 9

Sous-tâche 1: Nouveaux ponts

Nous pouvons y parvenir en ajoutant seulement deux arêtes : $5 \rightarrow 1$ et $2 \rightarrow 6$.



Cette représentation du réseau de ponts à l'aide de nœuds (= quartiers) et d'arêtes (= ponts) s'appelle un graphe. Ici, les arêtes ne peuvent être empruntées que dans un seul sens. Nous parlons donc d'un graphe orienté, dont les arêtes orientées sont appelées des arcs.

Pour cette tâche, la notion de « composante fortement connexe » (CFC, ou SCC en anglais) est utile. Une CFC est un ensemble de nœuds dans lequel nous pouvons atteindre n'importe quel nœud depuis n'importe quel autre nœud de l'ensemble. Au départ, les nœuds 4, 7 et 8 forment une CFC. Le but est de créer une seule grande CFC contenant tous les nœuds, en ajoutant le moins d'arcs possible.

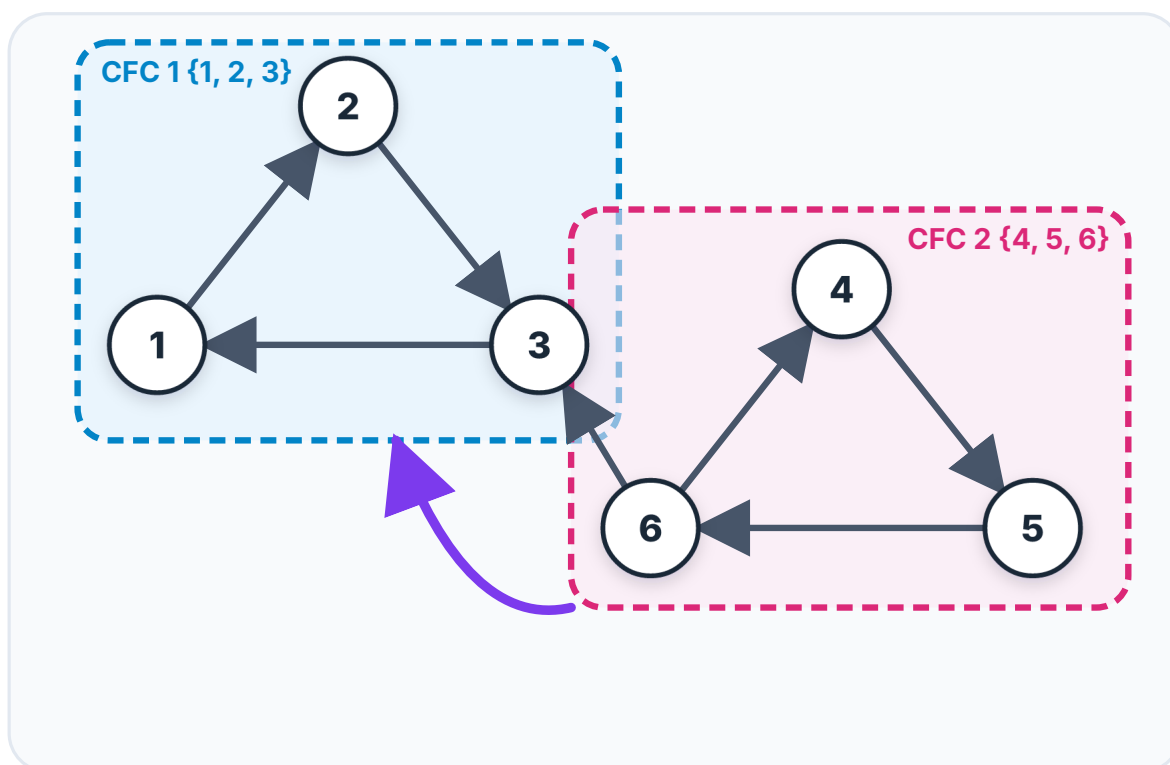
Nous devons ajouter un arc sortant à chaque nœud qui n'en possède pas encore, ainsi qu'un arc entrant à chaque nœud qui n'en possède pas encore. Les nœuds sans arc sortant sont initialement les nœuds 2 et 5. Les nœuds sans arc entrant sont initialement les nœuds 1 et 6. En les associant de manière à relier le petit « îlot » constitué de 1 et 2 au reste du graphe, nous pouvons créer une seule grande CFC.

Notons que cela ne suffit pas en général. Il existe des exemples où tous les nœuds ont des arcs entrants et sortants, mais où le graphe ne forme toujours pas une seule grande CFC. Le graphe de la partie 2 en est un contre-exemple.

Sous-tâche 2: Un nouveau pont

Il y a 9 façons d'ajouter un arc pour satisfaire la condition.

Comme dans la première partie, nous voulons créer une seule grande CFC. Le graphe possède actuellement deux petites CFC : l'une contient les nœuds 1, 2 et 3, l'autre les nœuds 4, 5 et 6. Comme nous pouvons atteindre tous les autres nœuds à l'intérieur d'une CFC, nous pouvons, en quelque sorte, considérer chaque CFC comme un seul nœud et remplacer tous ses nœuds par celui-ci. Le graphe devient alors très simple : il ne comporte plus que deux nœuds, un pour chaque CFC, avec un arc allant de la deuxième CFC vers la première.



Depuis un nœud de la deuxième CFC, nous pouvons déjà atteindre tous les autres nœuds du graphe. En revanche, il est impossible de passer d'un nœud de la première CFC à un nœud de la deuxième. Cela se voit aussi clairement dans notre graphe simplifié à deux nœuds.

Nous devons donc ajouter un arc allant de la première CFC vers la deuxième. Comme tous les nœuds d'une CFC sont accessibles les uns depuis les autres, le choix du nœud à l'intérieur de chaque CFC n'a pas d'importance. Nous pouvons donc ajouter un arc de n'importe quel nœud de la première CFC vers n'importe quel nœud de la deuxième. Nous avons trois choix dans la première CFC et trois dans la deuxième. Il y a donc $3 \cdot 3 = 9$ arcs possibles dont l'ajout permet de réunir le graphe en une seule grande CFC.



Atlantide

Préparation de la tâche

Traduction (de)

Traduction (fr)

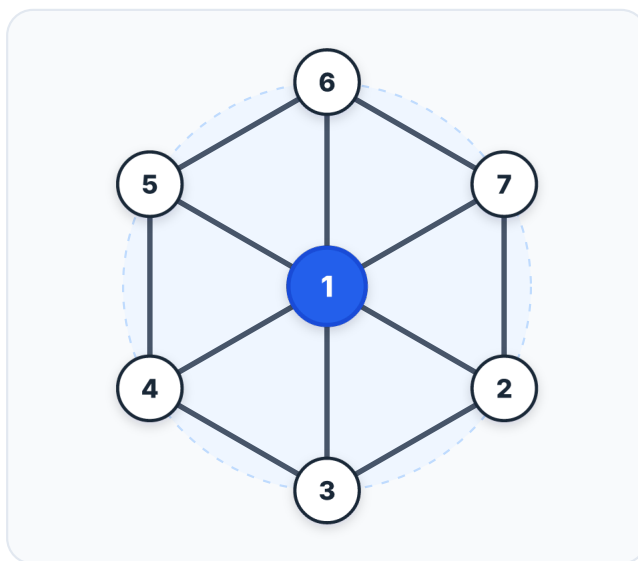
Auteur de la solution

Mark Neumann

Benjamin Schmid

Cheng Zhong

Benjamin Schmid



Partie 1: Le pays mythique de l'Atlantide possède un réseau ferroviaire particulier reliant ses n villes. Il est constitué d'un arrêt central, relié par une ligne ferroviaire directe à chacune des autres villes. Les $n - 1$ villes restantes forment un anneau de lignes ferroviaires autour de la ville centrale. Un exemple pour $n = 7$ est illustré. Formellement, il y a une ligne ferroviaire directe entre la ville 1 et chacune des autres villes. Il y a également une ligne ferroviaire directe de la ville x à la ville $x + 1$ pour tout $2 \leq x \leq n - 1$, ainsi qu'une ligne ferroviaire directe de la ville 2 à la ville n . Chaque ligne ferroviaire peut être empruntée dans les deux directions.

La souris Stofl aimerait parcourir l'Atlantide en partant d'une station et en voyageant le long du réseau ferroviaire. Cependant, pour que cela reste intéressant, il ne veut emprunter chaque ligne ferroviaire qu'au plus une fois. Quel est le nombre maximal de lignes ferroviaires que la souris Stofl peut emprunter consécutivement lors de son parcours de l'Atlantide, sans utiliser deux fois la même ligne, pour $n = 7$?

Un exemple de parcours valide que Stofl peut emprunter est $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 5$. Ce chemin est composé de 4 lignes ferroviaires.

☐ 7

☐ 8

☐ 9

☐ 10

☐ 11

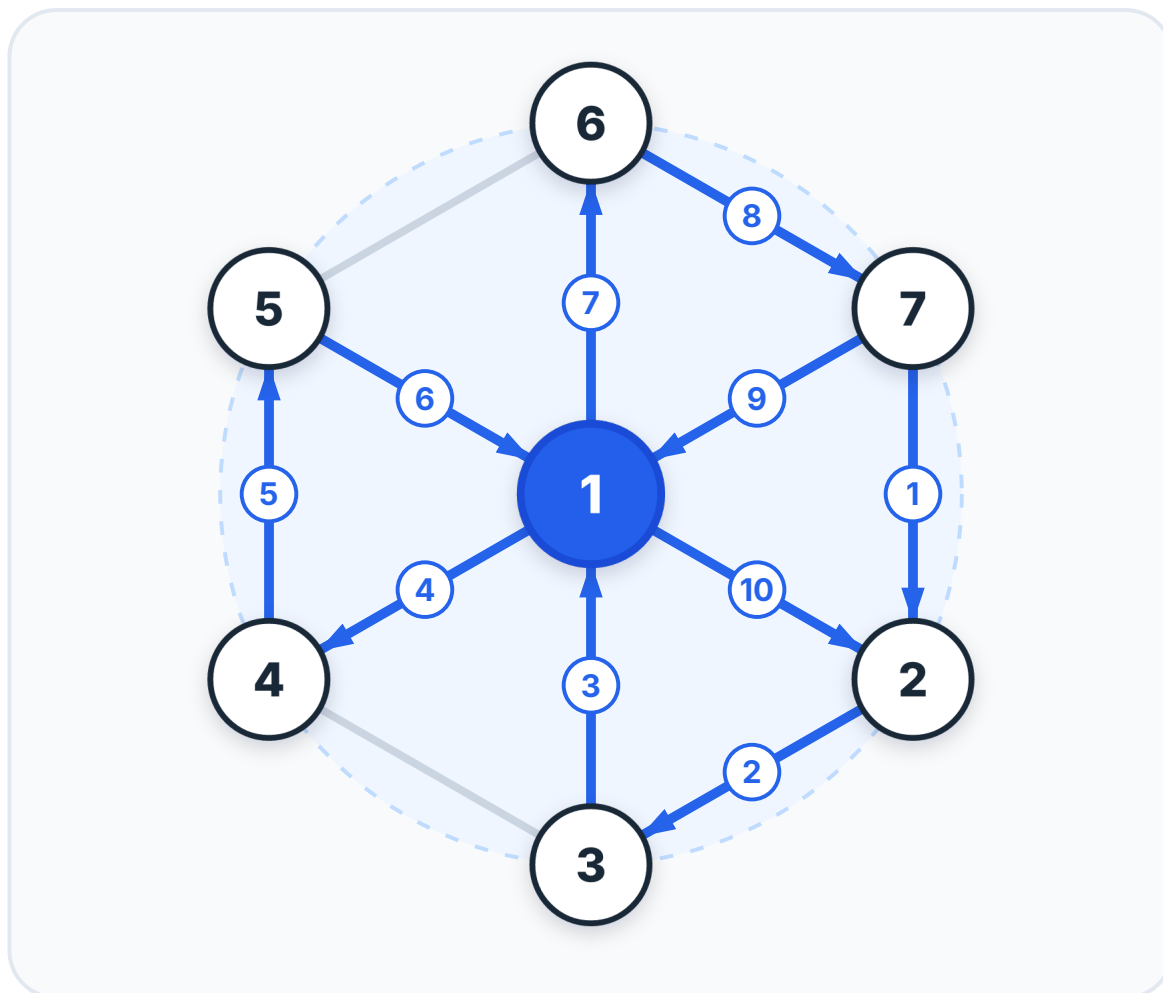
Partie 2: Supposons maintenant que l'Atlantide soit bien plus grande et que $n = 2027$. Quel est le plus long parcours que Stofl peut emprunter maintenant ?

.....



Sous-tâche 1: Petit pays

Le plus long parcours que nous pouvons trouver a une longueur de 10. Il en existe plusieurs ; en voici un exemple : $7 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow 1 \rightarrow 2$.



Cette représentation du réseau ferroviaire à l'aide de nœuds (= villes) et d'arêtes (= lignes ferroviaires) s'appelle un graphe.

Il existe plusieurs façons de se déplacer dans un graphe. La plus simple s'appelle une chaîne : nous passons simplement d'un nœud à l'un de ses voisins auquel il est relié par une arête. Nous pouvons passer plusieurs fois par chaque nœud et emprunter plusieurs fois chaque arête. Si nous imposons de ne visiter chaque nœud qu'au plus une fois, nous parlons d'une chaîne élémentaire. Dans notre cas, il n'y a pas de restriction sur les nœuds, mais nous ne pouvons emprunter chaque arête qu'au plus une fois. C'est ce que nous appelons une chaîne simple.

Le but est de trouver la plus longue chaîne simple possible dans un graphe. En général, il n'existe pas d'algorithme efficace pour la trouver, mais nous pouvons utiliser la forme particulière du graphe pour construire une solution. Ici, comme le graphe est petit, nous pouvons simplement essayer à la main différentes chaînes simples ou différentes manières de les construire.



Sous-tâche 2: Grand pays

Le plus long parcours a une longueur de 3040.

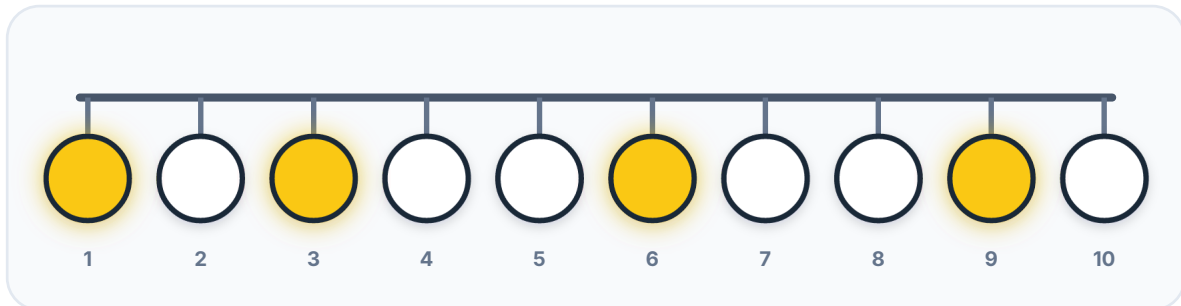
Comme le graphe est trop grand pour trouver un tel parcours à la main, nous devons repérer un motif qui se répète. En observant la solution de la partie 1, nous voyons que le parcours va le plus souvent du centre à un nœud extérieur, puis à un voisin, avant de revenir au centre. Ce motif est légèrement modifié au début et à la fin du parcours. Sans ces deux cas particuliers, notre parcours serait simplement constitué des triangles formés par ce motif.

Combien de tels triangles pouvons-nous placer sans qu'ils partagent d'arêtes ? Chaque triangle nécessite une arête extérieure, et nous pouvons utiliser une arête extérieure sur deux pour former un triangle. Chaque triangle ajoute 3 à la longueur de notre parcours. Avec n nœuds, nous avons $n - 1$ arêtes extérieures. Le motif habituel produit donc un parcours de longueur $\frac{n-1}{2} \cdot 3$.

En traitant le début et la fin de manière particulière, nous pouvons ajouter une arête supplémentaire à notre parcours. Sa longueur finale est donc $\frac{n-1}{2} \cdot 3 + 1$. Pour $n = 2027$, nous obtenons ainsi un parcours de longueur $\frac{2027-1}{2} \cdot 3 + 1 = 1013 \cdot 3 + 1 = 3040$.

LED

Préparation de la tâche	Mark Neumann
Traduction (de)	Benjamin Schmid
Traduction (fr)	Cheng Zhong
Auteur de la solution	Benjamin Schmid



Partie 1: Stofl est régisseur de scène et installe une rangée de 10 projecteurs LED pour un concert. Actuellement, certaines lumières brillent en jaune (ALLUMÉES) et d'autres sont blanches (ÉTEINTES).

Il veut allumer les 10 projecteurs, mais il y a un hic : le panneau de commande est câblé de manière étrange. Il ne peut appuyer que sur des boutons qui commandent deux lumières adjacentes exactement en même temps. Chaque fois qu'il appuie sur un bouton commandant deux lumières adjacentes, les états de ces deux lumières changent. (Si une lumière était ALLUMÉE, elle s'éteint ; si elle était ÉTEINTE, elle s'allume.)

La configuration initiale des 10 lumières vous est donnée sur l'image.

Quel est le nombre minimal de pressions de boutons que Stofl doit effectuer pour que les 10 lumières soient ALLUMÉES ?

- ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9

Partie 2: Stofl se rend compte qu'en raison du câblage étrange du panneau de commande, il existe peut-être des configurations initiales pour lesquelles il est impossible d'allumer les 10 lumières, quel que soit le nombre de boutons qu'il presse, mais il n'en est pas sûr.

Chacune des 10 lumières peut initialement être soit ALLUMÉE, soit ÉTEINTE, ce qui signifie qu'il y a de nombreuses configurations initiales possibles différentes.

Parmi toutes les configurations initiales possibles de ces 10 lumières, combien d'entre elles peuvent être entièrement ALLUMÉES à l'aide des boutons de lumières adjacentes de Stofl, après un certain nombre d'opérations, éventuellement nul ?

.....



Sous-tâche 1: Allumer les lumières

Il faut effectuer au moins 6 pressions de boutons.

Nous pouvons faire quelques observations :

- Inverser les états (ÉTEINTE, ALLUMÉE) donne (ALLUMÉE, ÉTEINTE), et inversement.
- Appuyer deux fois sur un bouton est inutile, car la deuxième pression annule la première. Cela reste vrai même si nous avons appuyé sur d'autres boutons entre les deux.
- La dernière pression doit concerner deux lumières dans l'état (ÉTEINTE, ÉTEINTE). Sinon, elle éteint forcément une lumière et nous n'avons pas terminé.

Pour déterminer le nombre minimal de pressions, nous pouvons maintenant parcourir les lumières de gauche à droite. Nous devons inverser l'état de la lumière 2, ce qui nous laisse deux possibilités : inverser les lumières 1 et 2, ou les lumières 2 et 3. Inverser les lumières 1 et 2 ne convient pas, car la lumière 1 serait alors ÉTEINTE et il ne resterait aucun bouton sur lequel appuyer pour la rallumer. Nous devons donc inverser les lumières 2 et 3. Il faut ensuite rallumer la lumière 3. Le même raisonnement ne laisse qu'une seule possibilité : inverser les lumières 3 et 4.

En parcourant toutes les lumières de cette manière, nous pouvons déterminer exactement quels boutons utiliser : ceux qui commandent les paires 2 et 3, 3 et 4, 5 et 6, 6 et 7, 8 et 9, ainsi que 9 et 10.

Sous-tâche 2: Configurations impossibles

Il y a 512 configurations à partir desquelles nous pouvons allumer toutes les lumières.

Il existe au total $2^{10} = 1024$ configurations possibles. Appuyer sur un bouton inverse toujours l'état d'exactly deux lumières. Le nombre total de lumières allumées reste donc identique, augmente de deux ou diminue de deux. En particulier, sa parité ne change pas. Ainsi, si le nombre de lumières initialement allumées est impair, nous ne pourrions jamais toutes les allumer. En revanche, s'il est pair, il est toujours possible de toutes les allumer en utilisant la procédure de la partie 1.

Grâce à cette observation, il nous suffit de compter les configurations qui ont un nombre pair de lumières allumées. Cela représente exactement la moitié de toutes les configurations, soit 512.



Cogros

Préparation de la tâche	Mark Neumann
Traduction (de)	Benjamin Schmid
Traduction (fr)	Cheng Zhong
Auteur de la solution	Benjamin Schmid

Partie 1: Dans le nouveau centre commercial Cogros, il y a 256 portes automatiques. Elles sont toutes actuellement ÉTEINTES, et chaque porte possède son propre interrupteur.

Vous recevez une information anonyme selon laquelle exactement un de ces 256 interrupteurs est complètement défectueux, mais vous ne savez pas lequel. Votre tâche est de le trouver !

Chaque minute, vous pouvez choisir d'effectuer exactement une des deux actions suivantes :

- Actionner un interrupteur : s'il s'agit de l'interrupteur défectueux, rien ne se passe. S'il s'agit d'un interrupteur qui fonctionne, il change l'état de la porte (d'ÉTEINTE à ALLUMÉE, ou d'ALLUMÉE à ÉTEINTE). Cependant, actionner l'interrupteur ne vous permet pas de voir la porte, vous ne connaîtrez donc pas immédiatement son nouvel état.
- Inspecter une porte : vous vous rendez auprès d'une porte précise pour la regarder et voir si elle est actuellement ALLUMÉE ou ÉTEINTE.

Quel est le nombre minimal de minutes dont vous avez besoin pour garantir que vous trouverez l'interrupteur défectueux, même dans le pire des cas (si vous êtes extrêmement malchanceux) ?

☐ 384 ☐ 255 ☐ 256 ☐ 510 ☐ 512

Partie 2: Après y avoir réfléchi un peu plus, vous vous rendez compte qu'il existe une troisième action que vous pouvez choisir d'effectuer. Comme les autres, cette action prend elle aussi exactement une minute :

- Consulter le compteur électrique : vous pouvez mesurer la consommation totale d'électricité de tout le centre commercial Cogros. Cela vous indique instantanément combien de portes au total sont actuellement ALLUMÉES dans tout le bâtiment.

En utilisant cette nouvelle action, en plus des deux actions initiales, quel est le nouveau nombre minimal de minutes dont vous avez besoin pour garantir que vous trouverez l'interrupteur défectueux ?

Note : si vous actionnez plusieurs interrupteurs avant d'effectuer une vérification, chaque interrupteur prend toujours une minute entière.

.....



Sous-tâche 1: Trouver la porte défectueuse

Il nous faut au moins 510 minutes.

Il est impossible d'inspecter plusieurs portes ou d'actionner plusieurs interrupteurs en même temps. Vérifier si un interrupteur donné est défectueux prend donc exactement deux minutes : une minute pour l'actionner et une minute pour vérifier si la porte est ALLUMÉE.

Une stratégie possible consiste simplement à vérifier tous les interrupteurs, ce qui prend $2 \cdot 256 = 512$ minutes. Nous pouvons toutefois apporter une petite amélioration : une fois que nous avons vérifié tous les interrupteurs sauf un, il est inutile de vérifier le dernier, puisque nous savons qu'exactly un interrupteur est défectueux. Cela nous fait gagner deux minutes, et il ne faut donc plus que 510 minutes. C'est la meilleure solution.

Sous-tâche 2: Consommation d'électricité

Il nous faut au moins 263 minutes.

La nouvelle action nous permet de vérifier plusieurs portes à la fois. Nous pouvons ainsi utiliser une stratégie appelée recherche dichotomique. Nous commençons par actionner exactement la moitié des interrupteurs, puis nous consultons le compteur électrique. Si le nombre de portes ALLUMÉES correspond au nombre d'interrupteurs actionnés, nous savons que tous ces interrupteurs fonctionnent correctement. L'interrupteur défectueux se trouve donc dans l'autre moitié. En revanche, si les nombres ne correspondent pas, nous savons que l'un des interrupteurs actionnés est défectueux. Nous pouvons donc éliminer l'autre moitié.

Après une telle étape, nous avons éliminé la moitié des interrupteurs. Nous pouvons alors appliquer récursivement la même stratégie à la moitié dans laquelle se trouve l'interrupteur défectueux. Lorsqu'il ne reste plus que deux interrupteurs possibles, nous en vérifions un et pouvons ainsi identifier celui qui est défectueux.

Nous pouvons calculer le nombre de minutes nécessaires à l'aide du tableau suivant.

Interrupteurs restants	Interrupteurs à vérifier	Minutes nécessaires
256	128	129
128	64	65
64	32	33
32	16	17
16	8	9
8	4	5
4	2	3
2	1	2

En additionnant les valeurs de la dernière colonne, nous obtenons $129 + 65 + 33 + 17 + 9 + 5 + 3 + 2 = 263$ minutes.